MAKÜ
BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ

KONU : LİSTELERE GİRİŞ, BISECTION YÖNTEMİ

Konu Başlıkları

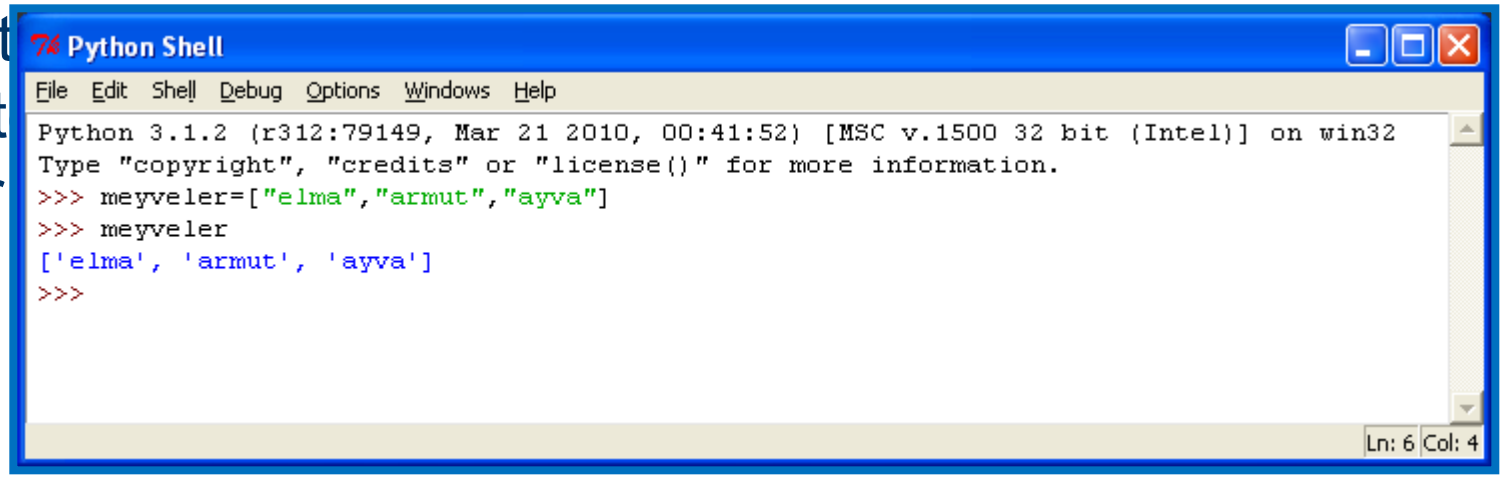
1. Liste Nedir?
    a. Liste tanımlama, indexleme, değer ekleme, değer çıkarma, del deyimi
    b. Liste sıralama, liste tersleme, değer arama
    c. Fonksiyonel Programlama Araçları
2. Bisection (İkiye Bölme) Nedir?
3. Örnekler
4. Kaynaklar

Başlıyoruz : Liste Nedir?

Listeler içinde birden fazla türde sayı ya da cümle sabitini barındıran birimlerdir.

Aslında listeler her türden python nesnesini(başka bir liste, tüp, sözlük, fonsiyon, sınıf) barındırabilir.

Bir list
da list
aralar

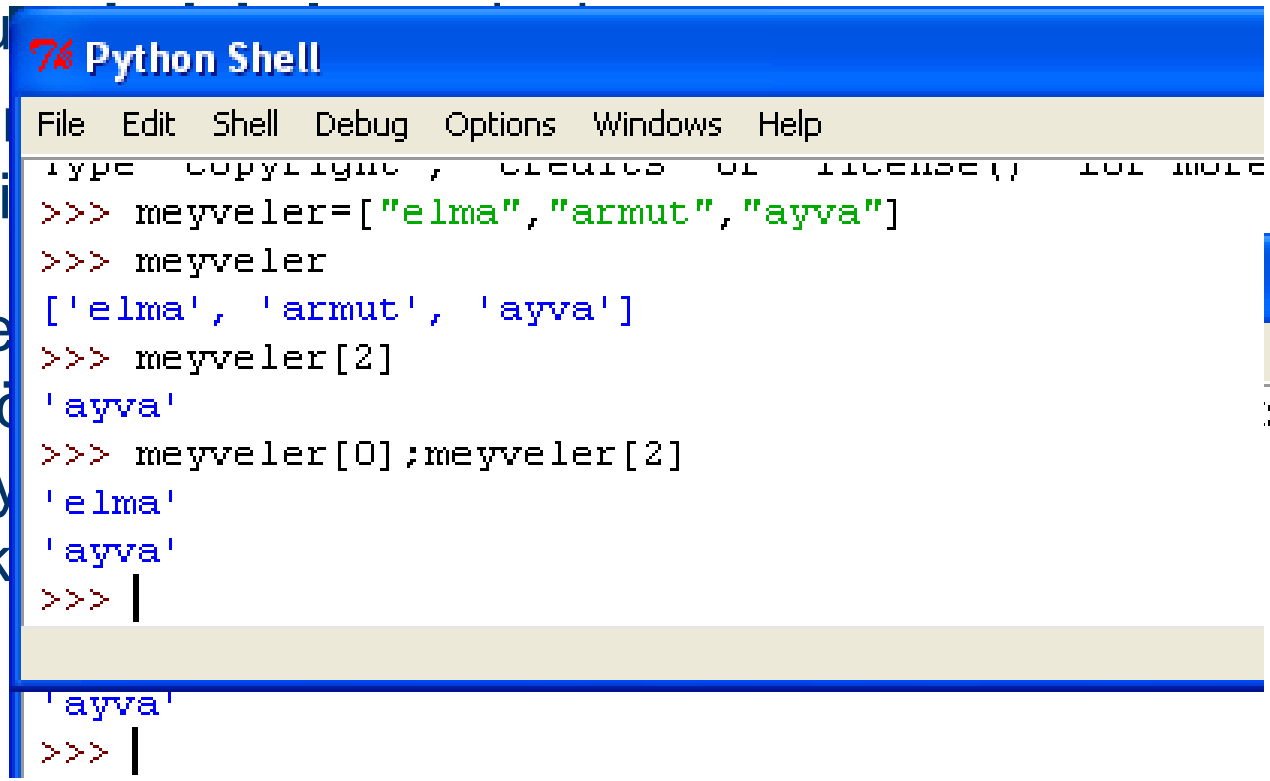


Listeler : Index Yazdırma

Listelerdeki her öğeye bulunduğu yere göre kolayca erişilebilir. Bu

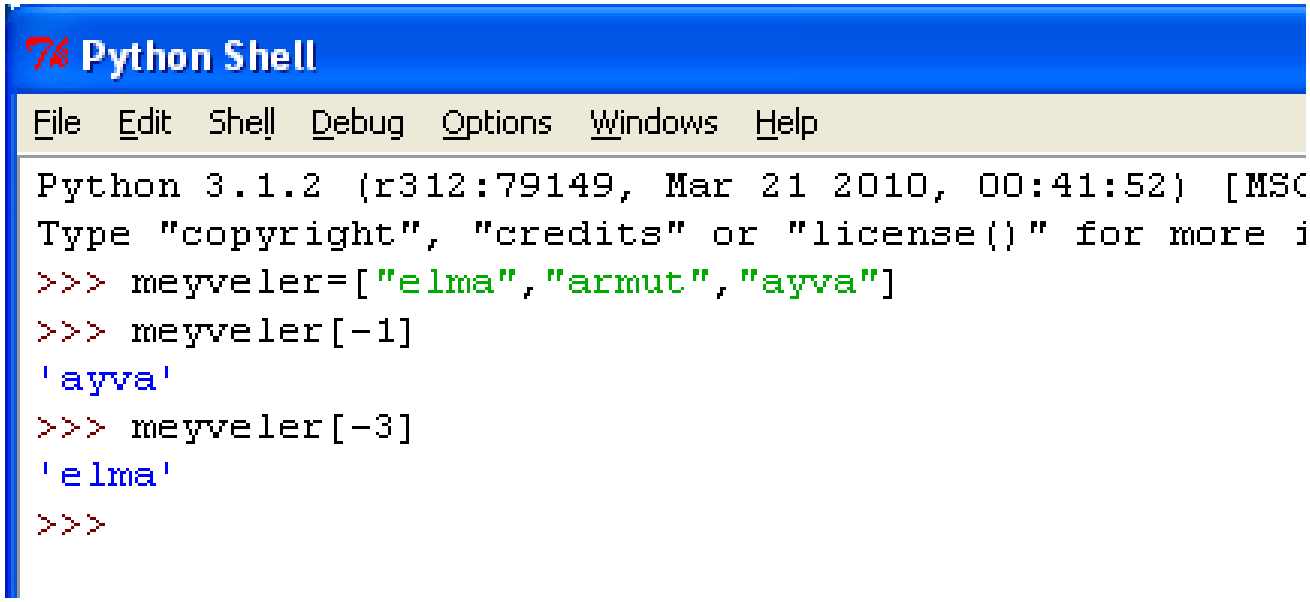
Örneğin az ö
üçüncü öğesi

Dikkat ettiyse
bize üçüncü ö
numaralamay
anda listedek



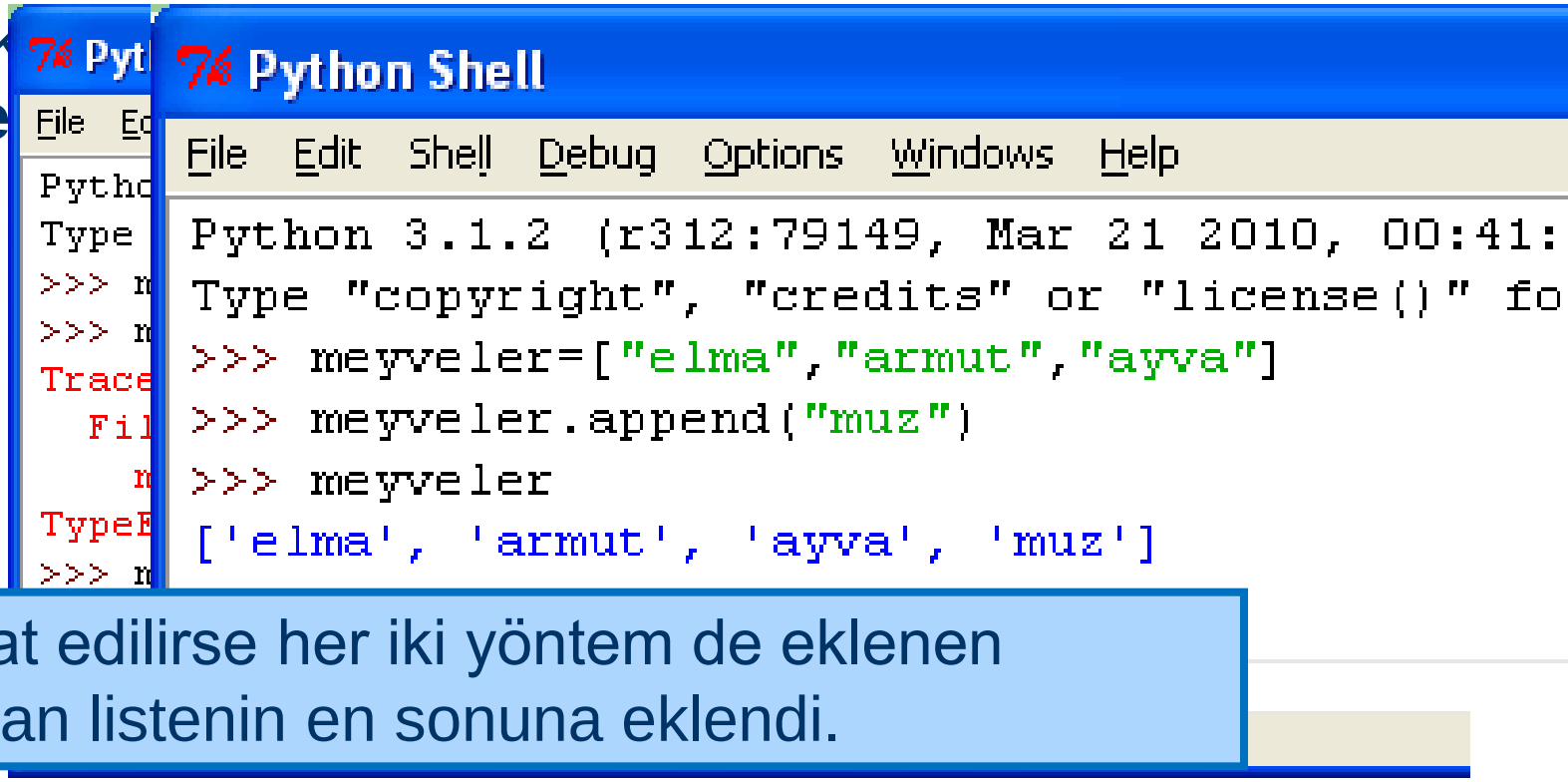
Listeler : Index (Sıra)

Listelerdeki her öğeye listenin başından erişim yapılabildiği gibi sonundan erişim yapılabilir. Örneğin ;



Listeler : append (Elemen Eklemek)

2. Yöntem daha sonradan eleman ekleyebilirsiniz. Bunun için ik
appe

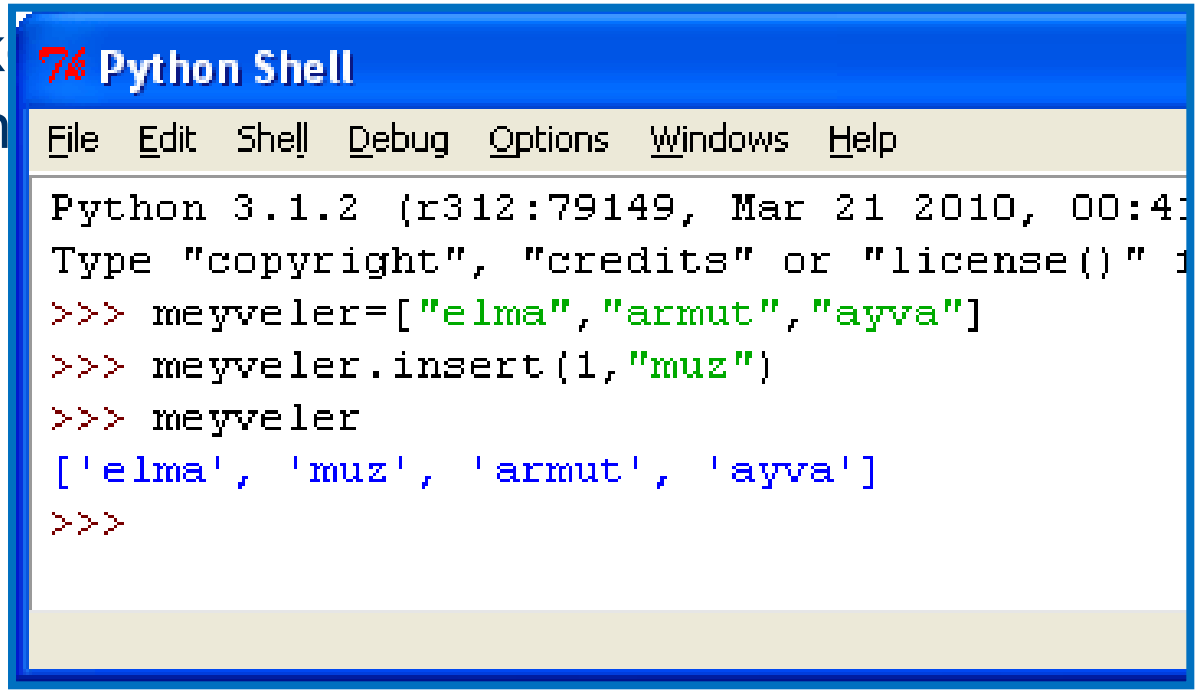


Dikkat edilirse her iki yöntem de eklenen eleman listenin en sonuna eklendi.

Listeler : insert (araya ekleme)

Bir önceki örnekte listenin sonuna nesne eklemeyi öğrendik.Ancak bazen listenin belirli bir indexine nesne eklememiz gerek
fonksiyonu kullan

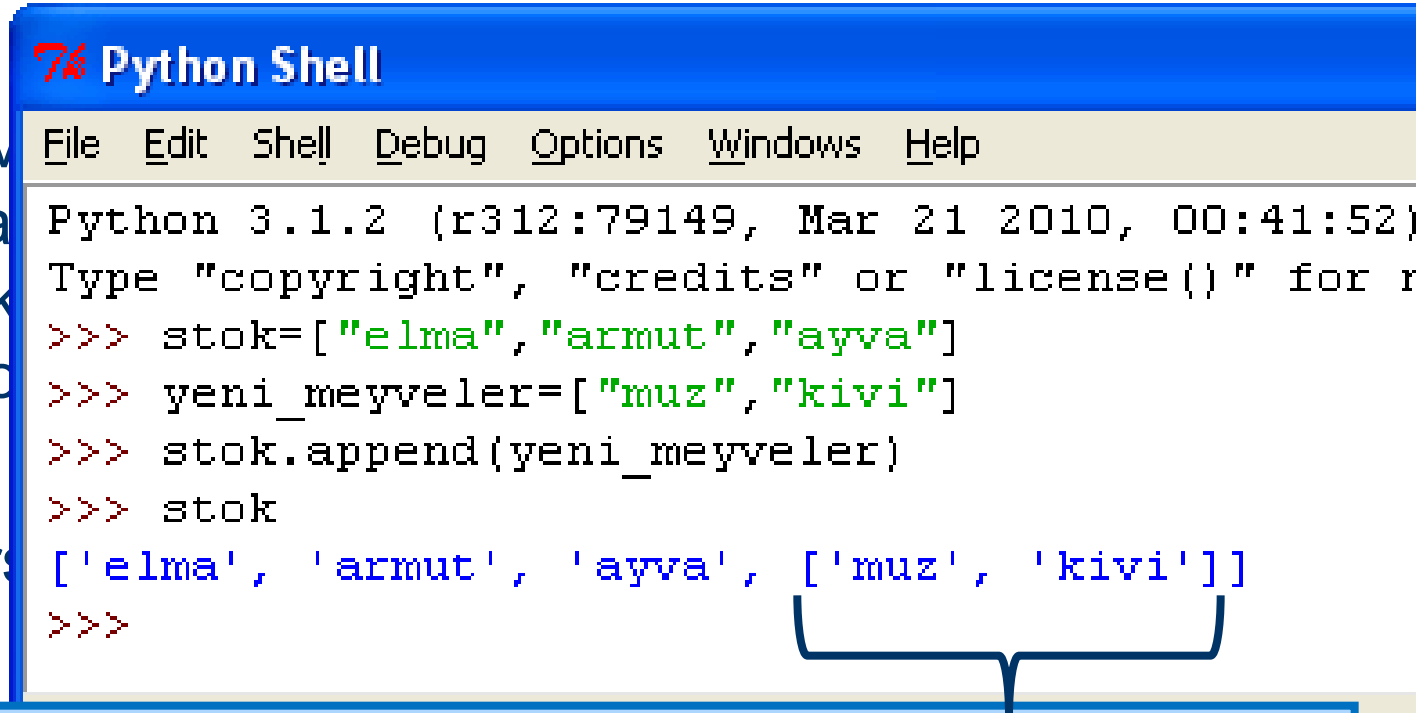
Örnek kullanımı ;



Listeler : extend (liste birleştirme)

İki listeyi
append v
kalmakta
Ancak ek
eleman c

Örnekler



Dikkat edilirse iki listeyi birleştirme sonucunda toplam 5 adet eleman olmalıydı. Ancak şu an 4 eleman var.

Listeler : extend (liste birleştirme)

Bu nedenle yeni e
birer nesne olarak

Doğru Kullanım :

```
>>> stok=["elma","armut","ayva"]
>>> yeni_meyveler=["muz","kivi"]
>>> stok.append(yeni_meyveler)
>>> stok
['elma', 'armut', 'ayva', ['muz', 'kivi']]
>>>
```

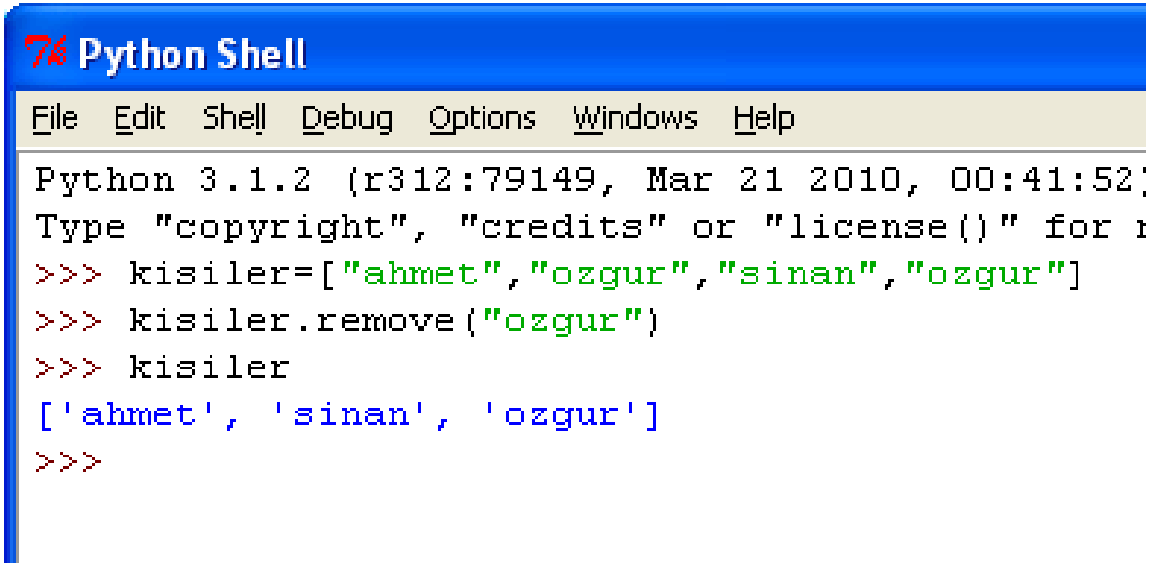
```
>>> stok=["elma","armut","ayva"]
>>> yeni_meyveler=["muz","kivi"]
>>> stok.extend(yeni_meyveler)
>>> stok
['elma', 'armut', 'ayva', 'muz', 'kivi']
```

extend işleminden sonra stok listesinin eleman sayısının 5 olduğuna dikkat!

Listeler : remove (nesne silme)

Listeden içeriği verilen nesnenin çıkartılmasını sağlar. Aynı nesneden liste içerisinde birden fazla olması durumunda ilk bulduğunu çıkartır.

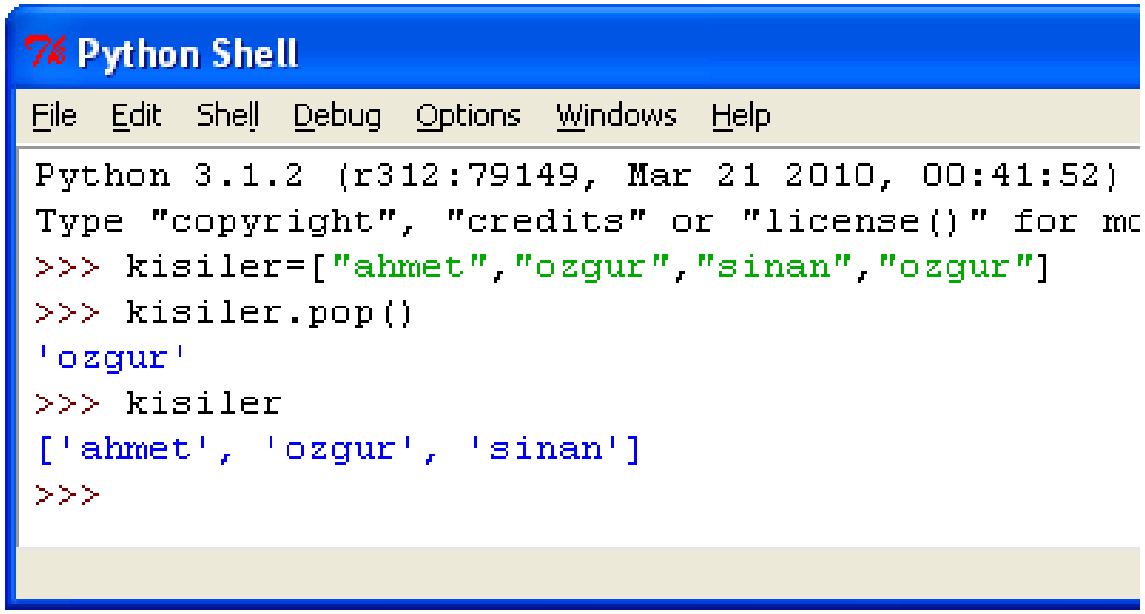
Örnek kullanımı ;



Listeler : pop (nesne silme)

Bir başka listeden eleman çıkarma yöntemi de **pop** komutu ile gerçekleştirilir. Default olarak **LIFO** mantığını işler. Listenin en son elemanını çıkartır ve çıkartılan eleman içeriği yazdırılır..

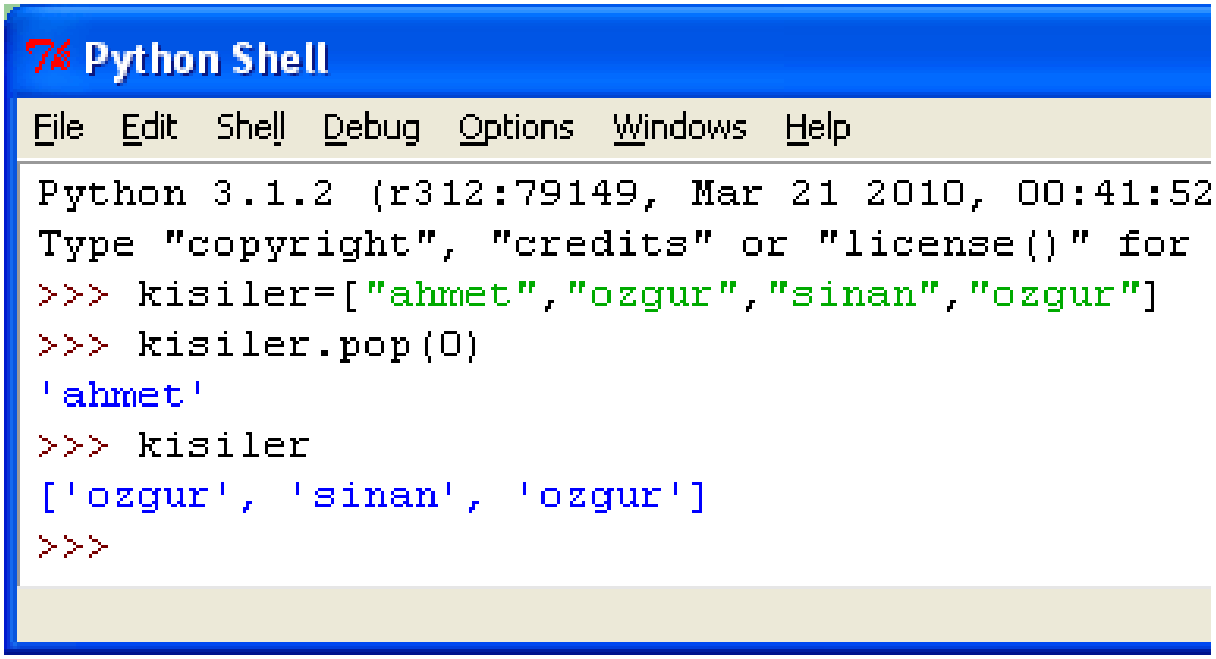
Örnek kullanımı ;



Listeler : pop (nesne silme)

Ancak istersek pop(?) bir index numarası belirterek çıkartma işlemide gerçekleştirilebilir.

Örnek kullanımı ;



Listeler : del (nesne ve liste silme)

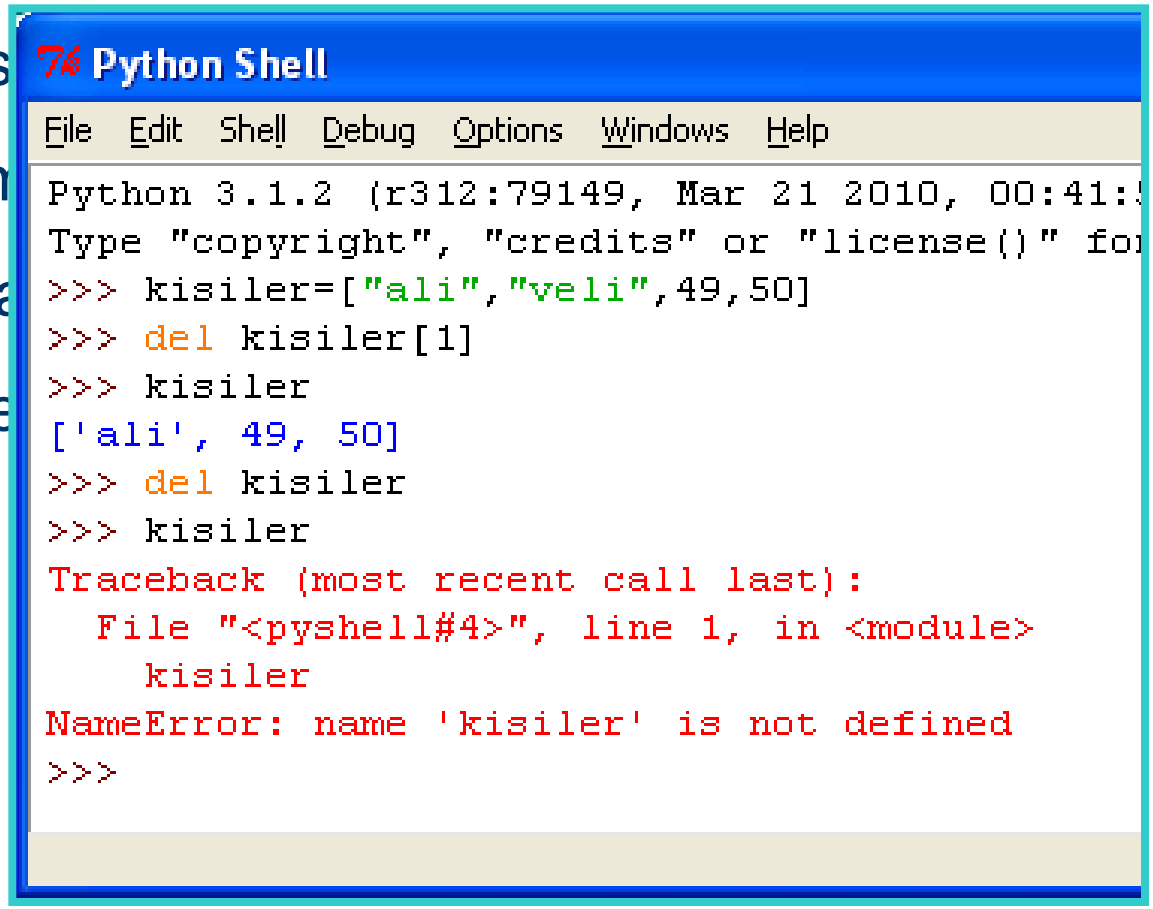
del() komutu ile lis

silinebilir. Aynı zan

Liste silindiği zama

tanımlanması gere

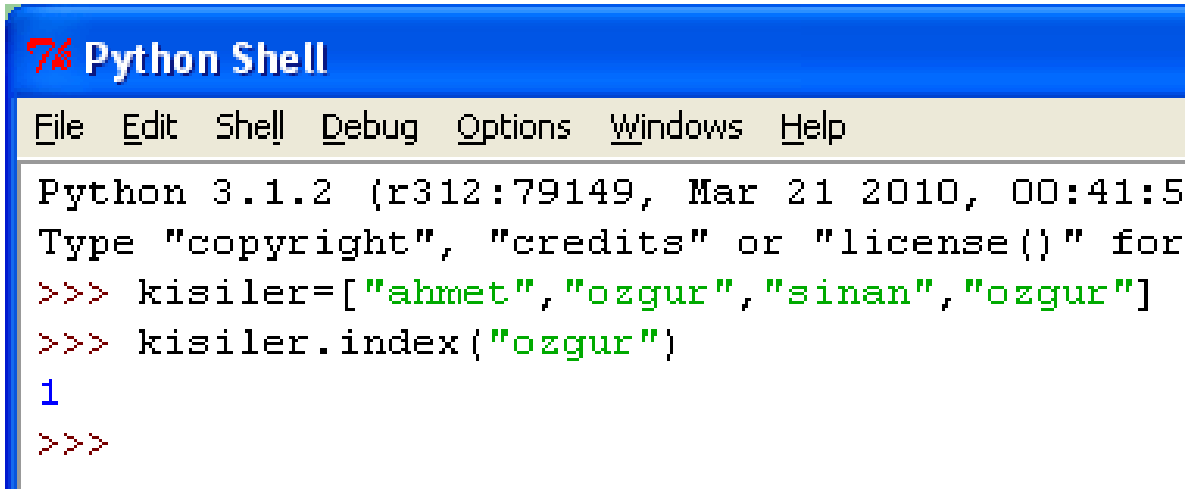
Örnek kullanımı ;



Listeler : index ()

Bazen bir elemanın liste içerisindeki index numarasını öğrenmek isteyebiliriz. Bu durumlarda **index()** komutu kullanılabilir. Remove komutu gibi ilk bulduğu sonucu işler.

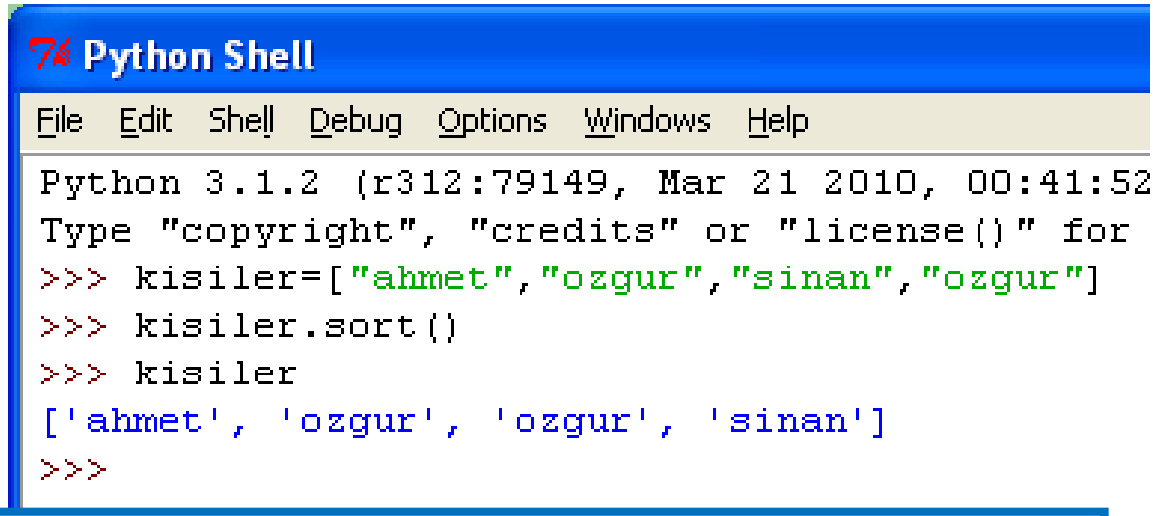
Örnek kullanımı ;



Listeler : sort ()

Karışık sıra ile oluşturmuş olduğumuz listeleri sıralamak python'da çok kolay…

Örnek kullanımı ;



Sıralamanın A→Z … 0 → 9 şeklinde olduğuna dikkat!

Listeler : reverse ()

Bir önceki örnekte sıralamayı A→Z … 0 → 9 şeklinde sort() komutu ile nasıl yapacağımızı öğrenmiştir. Bu sıralama işlemini terslemek için **reverse()** komutunu kullanabiliriz.

Örnek kullanımı ;

